

**AMENDMENTS TO THE CLAIMS**

Please amend the claims as follows.

1. (Currently Amended) A method for ~~checking page size dependency~~ processing a call comprising:

setting a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel;

generating an interposing library comprising a first modified interface, wherein the first modified interface is dependent on a native page size;

intercepting ~~[[a]] the call into [[a]] the kernel~~ by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on ~~[[a]] the non-native page size and wherein the kernel uses a native page size;~~

modifying the call by the interposing library using the first a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size;

sending the modified call to the kernel; ~~[[and]]~~

generating a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;

sending the response to the user-level application;

intercepting the response by the interposing library;

modifying the response to obtain a modified response, wherein the modified response is dependent on the non-native page size; and

sending the modified response to the user-level application.

2. - 5 (Cancelled)

6. (Currently Amended) The method of claim 1, further comprising:  
wherein generating the interposing library, wherein generating the interposing library  
comprises:  
searching a plurality of interfaces to determine which of the plurality of interfaces  
include the native page size; and  
modifying the plurality of interfaces that include the native page size to obtain a  
plurality of modified interfaces, wherein modifying the plurality of interfaces  
uses the non-native page size,  
wherein the modified interface is one of the plurality of modified interfaces.
7. (Cancelled)
8. (Currently Amended) A system for checking page size dependency comprising:  
a kernel, located in a kernel-level of the system, using a native page size;  
a user-level application located in a user-level of the system; and  
an interposing library located in the user-level configured to set a non-native page size to  
emulate and emulate [[a]] the non-native page size to the user-level application,  
wherein the interposing library emulates the non-native page size by modifying  
results from the kernel based on the non-native page size, wherein the results from  
the kernel are based on the native page size.
9. (Original) The system of claim 8, wherein the interposing library is further configured to modify  
a call dependent on the non-native page size from the user-level application to a call dependent  
on the native page size for the kernel.
10. (Original) The system of claim 8, wherein the interposing library uses a modified interface to  
emulate the non-native page size to the user-level application.
11. (Original) The system of claim 8, wherein the interposing library uses a modified interface to  
emulate the native page size to the kernel.

12. (Original) The system of claim 8, wherein the interposing library comprises a plurality of modified interfaces for emulating the non-native page size, wherein the plurality of modified interfaces are generated by searching a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size and modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces.
13. (Currently Amended) A computer system for checking page size dependency, comprising:
- a processor;
  - a memory;
  - a storage device;
  - a computer display; and
- software instructions stored in the memory for enabling the computer system under control of the processor, to:
- set a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel;
  - generate an interposing library comprising a first modified interface, wherein the first modified interface is dependent on a native page size;
  - intercept [[a]] the call into [[a]] the kernel by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on [[a]] the non-native page size and wherein the kernel uses a native page size;
  - modify the call by the interposing library using the first a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size;
  - send the modified call to the kernel; [[and]]
  - generate a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;
  - send the response to the user-level application;
  - intercept the response by the interposing library;

modify the response to obtain a modified response, wherein the modified response is dependent on the non-native page size; and  
send the modified response to the user-level application.

14. -17 (Cancelled)

18. (Currently Amended) The computer system of claim 13, wherein software instructions further comprise:

software instructions for generating the interposing library, wherein the software instructions comprise:

software instructions to search a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size; and  
software instructions to modify the plurality of interfaces that are dependent on the native page size to obtain a plurality of modified interfaces, wherein the modified interface is one of the plurality of modified interfaces.

19. (Cancelled)

20. (Currently Amended) A network system having a plurality of nodes, comprising:

a kernel, located in a kernel-level of the system, using a native page size;

a user-level application located in a user-level of the system; and

an interposing library located in the user-level configured to set a non-native page size to emulate and emulate [[a]] the non-native page size to the user-level application, wherein the interposing library emulates the non-native page size by modifying results from the kernel based on the non-native page size, wherein the results from the kernel are based on the native page size.

wherein the kernel executes on any node of the plurality of nodes,

wherein the user-level application executes on any node of the plurality of nodes,

wherein the interposing library executes on any node of the plurality of nodes,